Resource Allocation in Mobile Edge Learning

Duncan Mays

August, 2022

1 Overview and Motivation

With the proliferation of Internet of Things (IoT), it is expected that by 2027, 41 billion IoT devices will come online, generating an additional 800 zettabytes of data. The time-sensitive nature of this data is expected to force 90% of analytics to be performed at the edge to avoid latency of transmission to remote data centers as would be done in cloud computing. Thus, Multi-access Edge Computing MEC has emerged as a computing paradigm that can enable data processing at the edge (i.e., edge processing).

MEC consists of the utilization of dedicated edge infrastructure pieces built and owned by service providers to facilitate this data processing. This has the advantage of bringing compute resources close to the data which it must operate on, but the disadvantage of relying on companies and service providers for basic computing services that would normally be provided by hardware within the devices owned by consumers. Reliance on such an oligopoly can be avoided by tapping into the profuse compute resources in edge devices themselves such as smartphones, PCs, autonomous vehicles and other consumer-owned computers. Instead of depending on infrastructure that is owned by profit-motivated companies, consumer devices can share compute resources to accommodate each others' needs. This is Extreme Edge Computing (EEC), and differs from MEC in that the compute nodes have heterogeneous and uncertain characteristics.

Edge devices come in many forms, from the powerful Machine Learning (ML) acceleration hardware found in autonomous vehicles to the smallest IoT device operating on battery power. This heterogeneity in compute resources is a major issue for EEC, and forces orchestrators that wish to harness the distributed compute in these devices to have some way to assess their capabilities. Moreover, these edge devices are owned by consumers, who may try to use their devices for tasks other than what the EEC system has delegated to them. This causes the compute characteristics of these devices to be uncertain, as they can change over time when faced with load from other tasks. Moreover, consumer privacy is a paramount concern. Some consumers may not wish to share identifying characteristics of their devices with an orchestrator, and so scheduling decisions have to be made in a decentralized manner to preserve private information.

ML has found itself at the core of many technologies used by edge devices today. Object detection, speech recognition and text completion are all applications that depend on neural networks trained on huge corpusses of data. The data-intensive nature of training neural networks has motivated this training to be done at the edge, and thus Mobile Edge Learning (MEL) has been developed as a way for resource-constrained edge devices to collaboratively train a single model. Despite being resource-constrained individually, the collective power of such devices can be significantly profuse. The integration of these abundant yet underutilized computational resources with MEL provides a promising edge learning paradigm for a broad range of IoT and edge computing applications.

2 Challenges

Implementing algorithms in an MEL environment presents several problems which the developer must overcome. We now provide three of these issues associated with MEL, which are device heterogeneity, uncertainty in device capabilities and consumer privacy.

1. **Device Heterogeneity:** EEC entails the usage of consumer devices on the edge of the internet, tablets, smart devices, laptops. Such devices have varying compute and computation capabilities, as they possess different hardware, power needs and network connection quality. Heterogeneity of learner devices poses a resource allocation issue, as learners must be assigned a task no longer than the time given to solve it. Should all learners be assigned the same task, the cluster would be limited by its weakest learner, which is an unacceptable compromise.

2. **Uncertainty in Device Capabilities:** In a real MEL system, learner devices are owned by consumers and companies that may use them for other tasks throughout the training regime. This will interfere with the

training capabilities of affected learners through resource contention, and therefor poses another resource allocation issue. Variability in device capabilities, possibly due to contention with other tasks, can be modeled as uncertainty.

3. **Consumer Privacy:** Many consumers value their privacy, and so to improve the adoption of their product, companies must use algorithms that minimize the amount of data being sent over the network for analysis. In some cases, data exchanges are limited by laws such as the European Commission's General Data Privacy Regulation, which places stringent limits on data collection. Such concerns can extend to data on the compute capabilities of learner devices, as this information could be used to monitor user behavior or even identify devices.

3 Research Objectives

This work aims to solve the MEL learning problem by optimally allocating data between distributed learners. In single-task parallel learning, we compare the performance of a decentralized allocation method (DAB) against a centralized scheme (CSA). In multi-task parallel learning, we contrast three methods for data allocation in the presence of uncertainty, EOL minimizes the expected opportunity loss to ensure equitable allocation, MMET minimizes the training time required, and RSS completely ignores uncertainty. our four main objectives are summarized below:

1. **Objective 1:** In order to effectively allocate tasks between distributed learners with varying training capabilities, we must have a way to assess the training capabilities of learners. The objective is to be able to predict how much time a learner will need to train a given model on a given number of data samples. We evaluate our solution against industry-standard methods such as FLOPS.

2. **Objective 2:** To preserve privacy of learner device characteristics, we experiment with a decentralized allocation scheme where learners request an amount of data for training, rather than have it assigned to them by a centralized entity. Decentralized schemes are tolerant to learners that do not wish to participate fully, and only contribute a part of their total computational capabilities, as they can simply request less data than their maximum limit.

3. **Objective 3:** Distributed learners must synchronize in between updates to maintain consistency in the parameters being trained. This synchronization step can only occur once all learners have completed their update routine, and so it becomes important to ensure that every learner finishes their updates prior to a given training deadline. In this work we aim to allocate data such that all learners complete their local training regime within a global time constraint.

4. **Objective 4:** Learner capabilities will change over the course of a training regime due to resource contention with other tasks running on the learner.

It is impractical to continually measure training ability, and so we must model this variation as uncertainty in learner capabilities. A key objective in MEL is allocating data effectively in the presence of this uncertainty, which must be done by considering the expected delay of each training task, as calculated from the distribution of learner capabilities.

4 Contributions

1. **Contribution 1:** We propose Subset Benchmarking (SB) which is our method to estimate the training ability of each learner device. ML training tasks usually consist of the same process being repeated many times. Stochastic Gradient Descent (SGD) for instance, involves the same back-propegation calculation and parameter update being performed on each batch in a data set. SB works by running a subset of the training task and then extrapolating the time it took to predict the time required for the whole task. This is done for SGD by training over only a few batches from the dataset, and then scaling the amount of time it took to the whole dataset.

2. **Contribution 2:** Using SB, we can estimate the training capability of each learner, which we must use to allocate data between them to optimize some metric within a time constraint. In this work, we propose a method to determine the maximal amount of data that each learner is capable of processing in the given amount of time. These Data Upper Bounds (DUB)

then become constraints for the optimization calculation that allocates data between the learners.

3. **Contribution 3:** In this work, we consider the variations in learner capabilities over time due to resource contention with other tasks. We model these time variations as uncertainty in learner capabilities, and make calculations for DUB and optimization metrics as expected values across the distribution of learner capabilities.

5 Thesis Outline

This thesis is organized into the following chapters. Chapter 2 gives a review of the relevant literature on MEL, benchmarking, and data allocation in parallel machine learning. Chapter 3 provides a description of our method for decentralized data allocation using SB. Chapter 4 discusses multi-task parallel learning in the presence of uncertainty of learner capabilities. Chapter 5 concludes our work with a summary and gives directions for future research.